WE CLAIM:

1. A method of distributing software from a server to a client over a computer network, said method comprising:

downloading, from said server to said client, an applet including a caching and class loading mechanism; and

using said caching and class loading mechanism to retrieve and load additional applet modules on demand.

2. A method as in claim 1 wherein said client provides a particular applet execution environment, said downloading step comprises downloading an applet package that has been customized for said particular applet execution environment, and said using step includes retrieving and loading additional applet modules not specific to said particular applet execution environment.

3. A method as in claim 1 wherein said client provides a particular web browser, said downloading step comprises downloading an applet package that has been customized for said particular web browser, and said using step includes retrieving and loading additional applet modules not specific to said particular web browser.

4. A method as in claim 1 wherein said downloading step includes packaging said caching and class loading mechanism within a stream and downloading said stream in a single http transaction.

5. A method as in claim 4 wherein said stream includes an uncompressed archive file containing said caching and class loading mechanism.

1   6. A method as in claim 5 wherein said stream includes at least one digital

2 signature.

1   7. A method as in claim 1 wherein said using step includes verifying, under

2 control of code within said applet, at least one further applet module downloaded

3 on demand.

1   8. A method as in claim 1 further including constructing a cache map with

2 said caching and class loading mechanism.

1   9. A method as in claim 1 wherein said using step includes loading at least

2 further applet module to a persistent cache for subsequent use.

1   10. A method as in claim 1 wherein said additional applet modules are

2 organized into plural functional modules, and said using step comprises

3 downloading at least one of said functional modules on demand.

1   11. A method as in claim 1 wherein said client includes a local non-volatile

2 memory that persistently caches applet classes, and said using step includes

3 requesting an applet class from said server conditioned on determining whether said

4 applet class is already available in said non-volatile memory.

1   12. A method as in claim 1 wherein said client includes a local non-volatile

2 memory, and said using step includes determining whether a particular version of an

3 applet class is stored in said non-volatile memory, and requesting said version from

4 said server based on results of said determining step.

13. A method as in claim 1 wherein said using step includes persistently storing applet modules downloaded from said server to said client in a local non-volatile memory associated with said client.

14. A system for distributing software from a server to a client over a computer network, said system including a downloader that downloads, from said server to said client over said computer network, an applet including a caching and class loading mechanism, wherein said caching and class loading mechanism executes on said client to retrieve and load additional applet modules on demand.

15. A system as in claim 14 wherein said client provides a particular applet execution environment, said downloader downloads an applet package that has been customized for said particular applet execution environment, and said caching and loading mechanism retrieves and loads additional applet modules not specific to said particular applet execution environment.

16. A system as in claim 14 wherein said client provides a particular web browser, said downloader downloads an applet package that has been customized for said particular web browser, and said caching and loading mechanism retrieves and loads additional applet modules not specific to said particular web browser.

17. A system as in claim 14 wherein said downloader packages said caching and class loading mechanism within a stream and downloads said stream within a single http transaction.

18. A system as in claim 17 wherein said stream includes an uncompressed archive file containing said code representing caching and class loading mechanism.

24

1   19. A system as in claim 18 wherein said stream includes at least one digital

2   signature.

1   20. A system as in claim 14 wherein said caching and loading mechanism

2   verifies at least one further applet class downloaded on demand.

1   21. A system as in claim 14 wherein said caching and loading mechanism

2   maintains a cache map.

1   22. A system as in claim 14 wherein said client includes a persistent cache,

2   and said caching and loading mechanism loads at least further applet class to said

3   persistent cache for subsequent use.

1   23. A system as in claim 14 wherein said additional applet classes are

2   organized into functional modules, and said using caching and loading mechanism

3   downloads said functional modules on demand.

1   24. A system as in claim 14 wherein said client includes a local non-volatile

2   memory that persistently caches applet classes, and said caching and downloading

3   mechanism requests an applet class from said server conditioned on determining

4   whether said applet class is already available in said non-volatile memory.

1   25. A system as in claim 14 wherein said client includes a local non-volatile

2   memory, and said caching and loading mechanism determines whether a particular

3   version of an applet class is stored in said non-volatile memory, and requests said

4   version from said server based on results of said determining step.

1   26. A system as in claim 14 wherein said client includes a local non-volatile

2 memory associated therewith, and said caching and loading mechanism persistently

3 stores applet classes downloaded from said server to said client.

1   27. An applet comprising:

2   a first stream including a platform-dependent caching mechanism; and

3   at least one further stream including at least one platform-independent

4 functional module,

5   wherein said platform-dependent caching mechanism is used to cache and

6 load said platform-independent functional module.

1   28. A method of downloading an applet comprising:

2   (a) downloading a first stream including a platform-dependent caching and

3 loading mechanism; and

4   (b) downloading and caching, using said platform-dependent caching and

5 loading mechanism downloaded by step (b), at least one further stream including at

6 least one platform-independent functional module.

1   29. A system for downloading an applet comprising an applet environment

2 that receives, verifies and begins executing a first stream including a platform-

3 dependent caching and loading mechanism, wherein said applet environment, under

4 control of said platform-dependent caching and loading mechanism, requests,

5 receives and persistently caches at least one further stream including at least one

6 platform-independent functional module.

1   30. An applet-based execution model comprising:

2   initially executing downloaded applet classes that maintain a persistent cache

3   and request additional applet classes that are not in the persistent cache; and

4   subsequently executing said additional applet classes downloaded in

5   response to said additional applet class requests.

1   31. The applet-based execution model of claim 30 wherein said initially

2   executing step includes conditioning said requests on whether said additional applet

3   classes are in said persistent cache.